

8. HTTP POST Request

Write a Ruby script that makes an HTTP POST request to a specified URL with optional body parameters and prints the response.

- Method: `post_request`
 - Parameters:
 - `url`
 - `body_params`

```
(imn@hbtn-lab) - [.../scripting_cyber/0x00-ruby_scripting]
└─$ cat 8-main.rb
require_relative '8-post'

url = 'https://jsonplaceholder.typicode.com/posts'
body_params = { title: 'Try to Post', body: "Okay that's good", userId: 11,
id: 101 }
post_request(url, body_params)
```

```
(imn@hbtn-lab) - [.../scripting_cyber/0x00-ruby_scripting]
└─$ ruby 8-main.rb
Response status: 201 Created
Response body:
{
  "title": "Try to Post",
  "body": "Okay that's good",
  "userId": 11,
  "id": 101
}
```

File: `8-post.rb`

```
require 'net/http'
require 'uri'
require 'json'

def post_request(url, body_params = {})
  uri = URI.parse(url)
```

```

# Create an HTTP POST request
request = Net::HTTP::Post.new(uri)
request.content_type = 'application/json'
request.body = body_params.to_json unless body_params.empty?

# Initialize the HTTP session
response = Net::HTTP.start(uri.hostname, uri.port, use_ssl: uri.scheme ==
'https') do |http|
  http.request(request)
end

# Print response status and body
puts "Response status: #{response.code} #{response.message}"
puts "Response body:"
puts JSON.pretty_generate(JSON.parse(response.body))
end

```

Explanation

1. Required Libraries:

- `net/http`: Handles HTTP requests and responses.
- `uri`: Parses the URL into its components (scheme, host, path, etc.).
- `json`: Converts Ruby objects to JSON format and parses JSON responses.

2. Parse the URL:

```
uri = URI.parse(url)
```

Converts the `url` string into a URI object.

3. Create POST Request:

```

request = Net::HTTP::Post.new(uri)
request.content_type = 'application/json'
request.body = body_params.to_json unless body_params.empty?

```

- Sets the content type to JSON.
- Converts `body_params` (a Ruby hash) into a JSON string if it's not empty.

4. Start HTTP Session:

```

response = Net::HTTP.start(uri.hostname, uri.port, use_ssl: uri.scheme ==
'https') do |http|
  http.request(request)
end

```

- Starts an HTTP session.
- Automatically uses SSL if the URL scheme is `https`.

5. Print Response:

- Prints the status code and message (e.g., `201 Created`).
- Formats and displays the response body as pretty JSON.

Test Cases

Case 1: POST Request with Body Parameters

Command:

```
url = 'https://jsonplaceholder.typicode.com/posts'
body_params = { title: 'Try to Post', body: "Okay that's good", userId: 11,
id: 101 }
post_request(url, body_params)
```

Output:

```
Response status: 201 Created
Response body:
{
  "title": "Try to Post",
  "body": "Okay that's good",
  "userId": 11,
  "id": 101
}
```

Case 2: POST Request Without Body Parameters

Command:

```
url = 'https://jsonplaceholder.typicode.com/posts'
post_request(url)
```

Output:

```
Response status: 201 Created
Response body:
{
  "id": 101
}
```

Notes

1. The `201 Created` status is returned by the JSONPlaceholder API when a resource is successfully created.
2. Ensure the URL you're testing supports POST requests.
3. Modify `body_params` based on the API's expected input format.